

BioXSD

an XML format for basic bioinformatics data

The Bioinformatics Lab SS 2013
May 14th, 2013
Matúš Kalaš

BioXSD: an XML format for basic bioinformatics data

Abstract:

As an alternative to textual, binary, and RDF formats, XML has certain advantages in a number of usage scenarios (e.g. in object-oriented programming, or with Web services).

We have developed BioXSD as the XML alternative for sequence data, including alignments and feature records (Kalaš et al. 2010, Gundersen et al. 2011).

This presentation gives a quick overview of the main advantages of BioXSD.

an XML **format** for basic bioinformatics data

From the title, let's first focus on the word FORMAT.

What is a "format" ?

Examples of formats: PDF, .tex, JPEG, a Word document, MP3, DVD, FASTA format, GFF, SAM and BAM, XML, HTML, .tar.gz, text file and binary file, ...

One way of classifying different formats - especially applicable to formats of bioinformatics data – is by their structure into:

- linear or plain (such as plain-textual)
- tabular (2D; Tab Separated Values, TSV)
- XML (tree)
- RDF (oriented graph; RDF stands for Resource Description Framework)

Another basic classification would be into textual and binary formats (or files).

But it isn't so simple with format, as format is a pretty generic concept. There are many levels of formats, when data in one format can in addition be for example "wrapped", "serialised", "compressed", or "encoded" in another format, and that one can again be somehow put into yet another format and so on 😊 Can you come up with typical examples of this?

When I have a **format** ...

How can I parse it?

How can I validate it?

How can I write it?

For a software developer or a computational biologist, the main questions about a given format are the following:

How can it be parsed?

That is, how can I read it by my program into my variables inside the memory?

How can it be validated?

That is, how can I check whether the given piece of data (file) is formatted correctly, with respect to the 'definition' of the given format?

How can it be written?

That is, how can I write ("format") my data into the given format (when writing it into a file or a message or a database)?

an XML format for basic bioinformatics data

From the title, let's now focus on the XML.

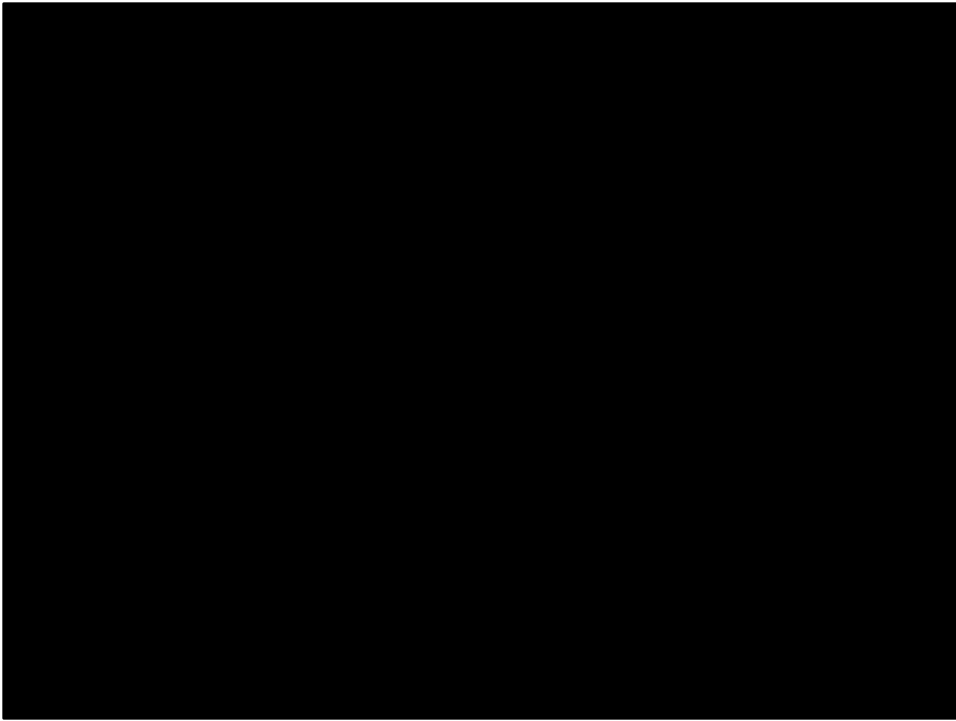
How can I parse it? Validate it? Write it?

What is the purpose of an XML Schema?

XML Schema is a language for defining a concrete XML format in a machine-understandable way.

What is the advantage of that?

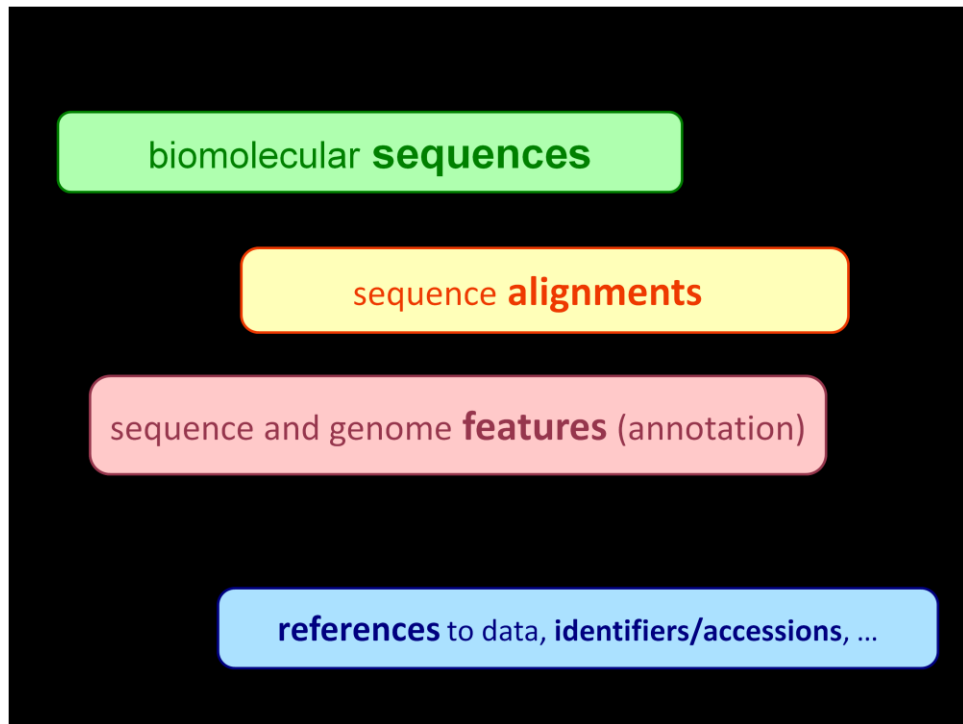
Get back to the 3 main questions to find the answer. (Isn't it psychedelic to find an answer among questions? 😊)



XML and XSD ... (see notes at the previous slide)

an XML format for basic bioinformatics data

Thirdly and lastly, let's say what the "basic bioinformatics data" from the title is.



By basic bioinformatics data we mean sequences, alignments, and their features (annotation).

Additionally for metadata (annotation), links to various resources such as databases, articles, taxonomies or ontologies (using some identifiers) are of great importance.

BioXSD

an XML format for basic bioinformatics data

Matúš Kalaš
Jan Christian Bryne*
Armin Töpfer**
Pål Puntervoll
Inge Jonassen

CBU, Uni Bergen

Edita Karosiene
Kristoffer Rapacki

CBS, DTU, Greater Copenhagen

Jon Ison

EBI, EMBL, Hinxton

Alexandre Joseph
Christophe Blanchet

IBCP, CNRS, Lyon

Steve Pettifer

University of Manchester

* now Oslo University Hospital

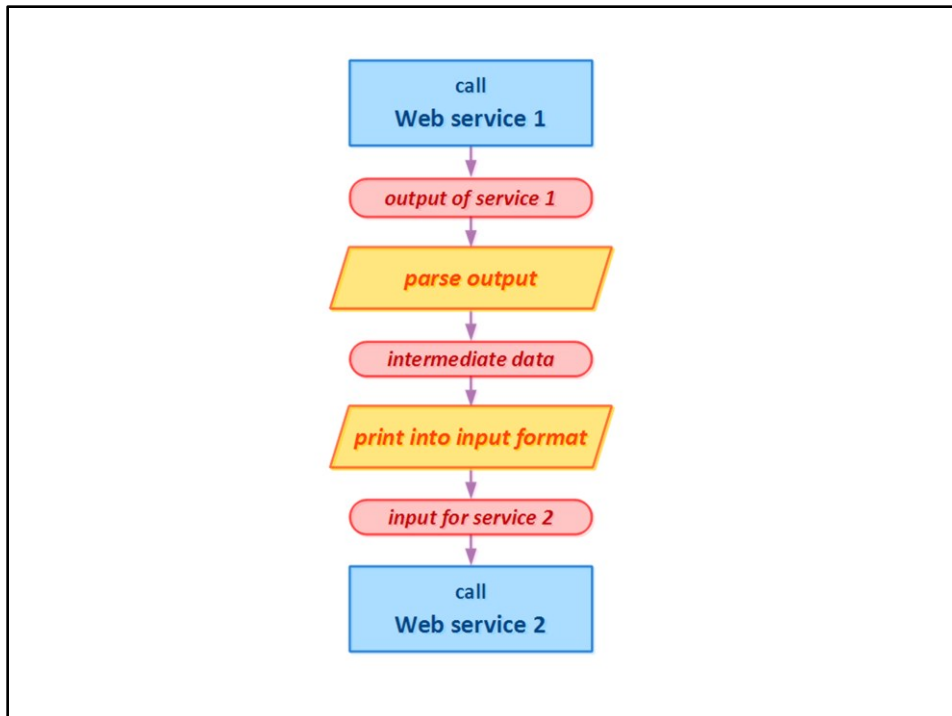
** now ETH Zürich in Basel

Now we can finally start with the BioXSD.

BioXSD is a community initiative organised at the University of Bergen in close collaboration with the Danish Technical University, and a few other collaborators. It is open towards more contribution from the broad bioinformatics community!

1. Common format

Point number one

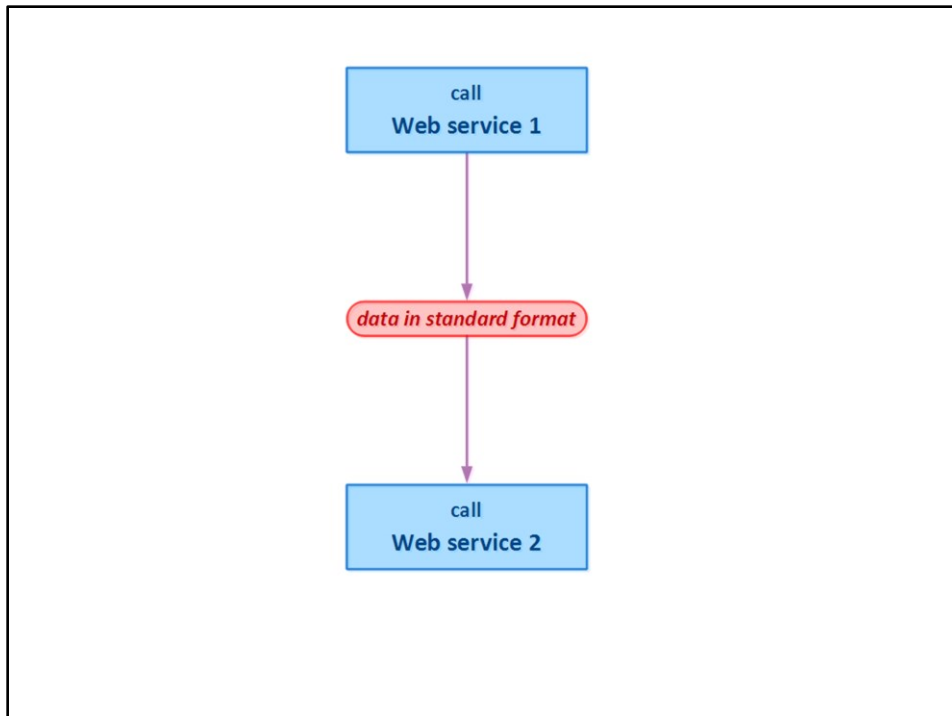


Before a common format is used by the 2 services/tools...

blue: tools/services

red: data

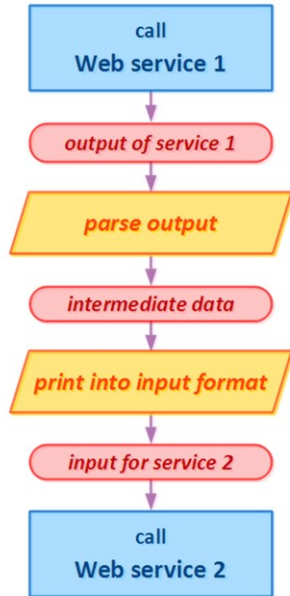
yellow: scripts, shims, the most favourite work of many bioinformaticians
(I loooove writing parsers, too☺)



After a common format is used by the 2 tools/services...

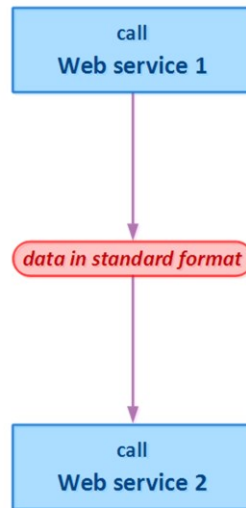
(or we can see it rather in form of an ad banner on the following slides😊)

Before



Before

After



After

/1. Common format

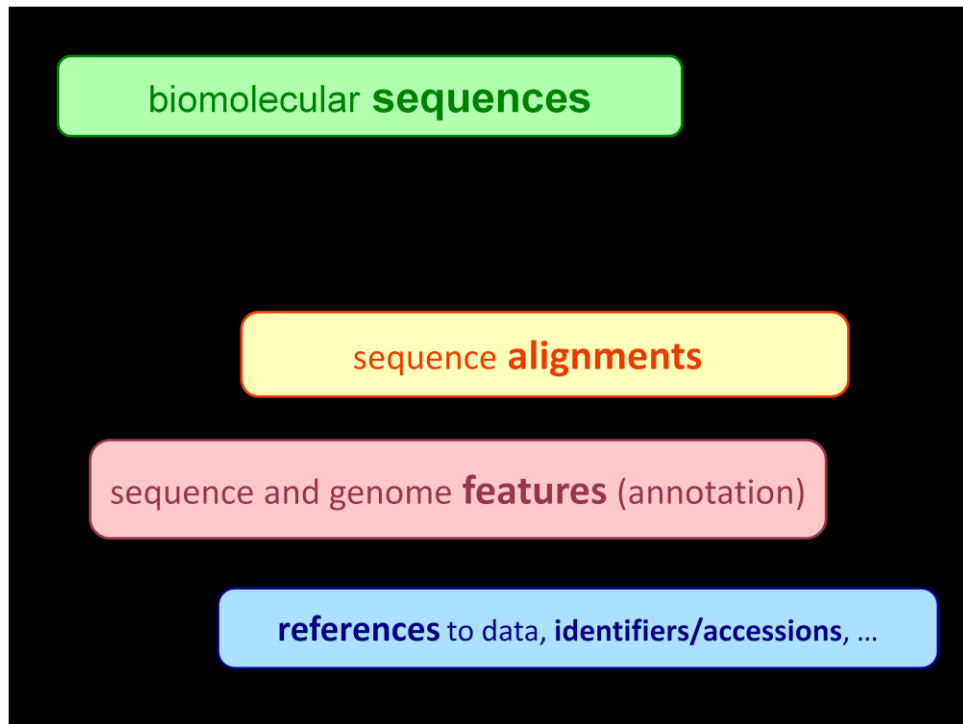
Advantage: smooth workflows

End of point number one

Obviously, the advantage when multiple tools/services use the same format is that they can easily be combined together in programs/scripts/workflows.

2. Rich format

Point number two



BioXSD is a rich format, but only within its well-defined and well-limited scope, filling the gap between other specialised XML formats

An example from the pocket of “biomolecular sequences” follows...

```

>sp|P43353|AL3B1_HUMAN Aldehyde dehydrogenase family 3 member B1 OS=Homo
sapiens GN=ALDH3B1 PE=1 SV=1
MDPLGDTLRRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQG
VLGGPQETGQLEHRFDYIFFTGSPRVGKIVMTAAAKHLTPVTLELGGKNPCYVDDNCDPQTVANRVAV
EPVMQEEIFGPILPIVENVQSLDEAIEFINRREKPLALYAFSNSSQVVKRVLTQTSSGGFCGNDGFMHMTLA

>AL3B1_HUMAN P43353 ALDEHYDE DEHYDROGENASE 3B1 (EC 1.2.1.5). - Homo sapiens
(Human).
MDPLGDTLRRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQG
RVGKIVMTAAAKHLTPVTLELGGKNPCYVDDNCDPQTVANRVAVFRYFNAGQTCVAPDYVLCSPEMQ
QTSSGGFCGNDGFMHMTLASLPFGGVGASGMGRYHGKFSFDTFSHHRACLLRSPGMEKLNALRYPPC

>gi|4502043|ref|NP_000685.1| aldehyde dehydrogenase family 3 member B1 isoform a
[Homo sapiens]
MDPLGDTLRRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQG
PRVGKIVMTAAAKHLTPVTLELGGKNPCYVDDNCDPQTVANRVAVFRYFNAGQTCVAPDYVLCSPEM
VLTQTSSGGFCGNDGFMHMTLASLPFGGVGASGMGRYHGKFSFDTFSHHRACLLRSPGMEKLNALRY

>sp_ac|P43353 \ID= AL3B1_HUMAN \DE="Aldehyde dehydrogenase family 3 member B1
(Aldehyde dehydrogenase 7)" \NCBITAXID=9606
MDPLGDTLRRRLREAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQG
LALRNLRAWMKDERVPKNLATQLDSAFIRKEPFGVLIIAPWNYPLNLTPLVGALAAGNCVVLKPSEI

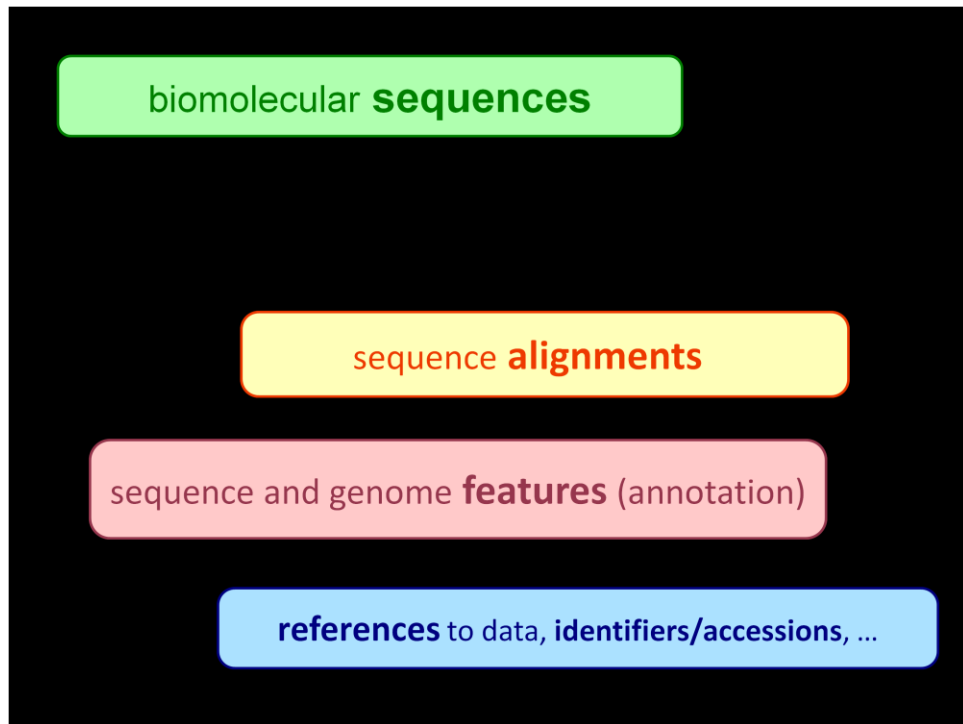
```

Different FASTA formats with different “deflines” i.e. headers i.e. metadata formats

Sequence record in BioXSD

```
<mySequence xsi:type="bx:GeneralAminoacidSequenceRecord">
  <bx:sequence>MDPLGDTLRRLEAFHAGRTRPAEFRAAQLQGLGRFLQENKQLLHDALAQDLHKSAFESEVSEVAISQGEVTLALRN
  YVDQSCFAVVLGGPQETGQLLEHRFDYIFFTGSPRVGKIVMTAAAKHLTPVTELGKKNPCYVDDNCDPQTVANRVAVFRYFNAGQT
  MEPVMQEEIFGPILPIVNVQSLDEAIEFINRREKPLALYAFSNSSQVVKRVLTQTSSGGFCGNDGFMHMTLASLPFGGVGASGMGRYH
  MEKLNALRYPPQSPRRLRMLLVAMEAQGCSTLL</bx:sequence>
  <bx:species
    dbName="NCBI Taxonomy"
    accession="9606"
    entryUri="http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=9606"
    speciesName="Human"
  />
  <bx:reference
    dbName="UniProt"
    accession="P43353"
    entryUri="http://www.uniprot.org/uniprot/P43353"
    sequenceVersion="1"
    variantAccession="P43353-1"
  />
  <bx:name>Aldehyde dehydrogenase family 3 member B1 (ALDH3B1)</bx:name>
</mySequence>
```

BioXSD represents *all* the metadata in an orderly and machine-friendly manner.



This was an example from the “biomolecular sequences” bit.

Another set of examples certainly interesting to mention would be from the complex bit, the “sequence and genome features”. BioXSD can represent feature records such as those formatted in GFF, BED, WIG, or SAM/BAM tabular formats, and such as those in proprietary formats of particular databases or outputs of tools (for example UniProt or ELMdb records, and BLAST or PredictProtein outputs)

/2. Rich format

Advantage: one format fits all

intermediate format for conversions

End of point 2.

(If you haven't before, note the geeky and boring XML "joke" 😊)

Rich format

=>

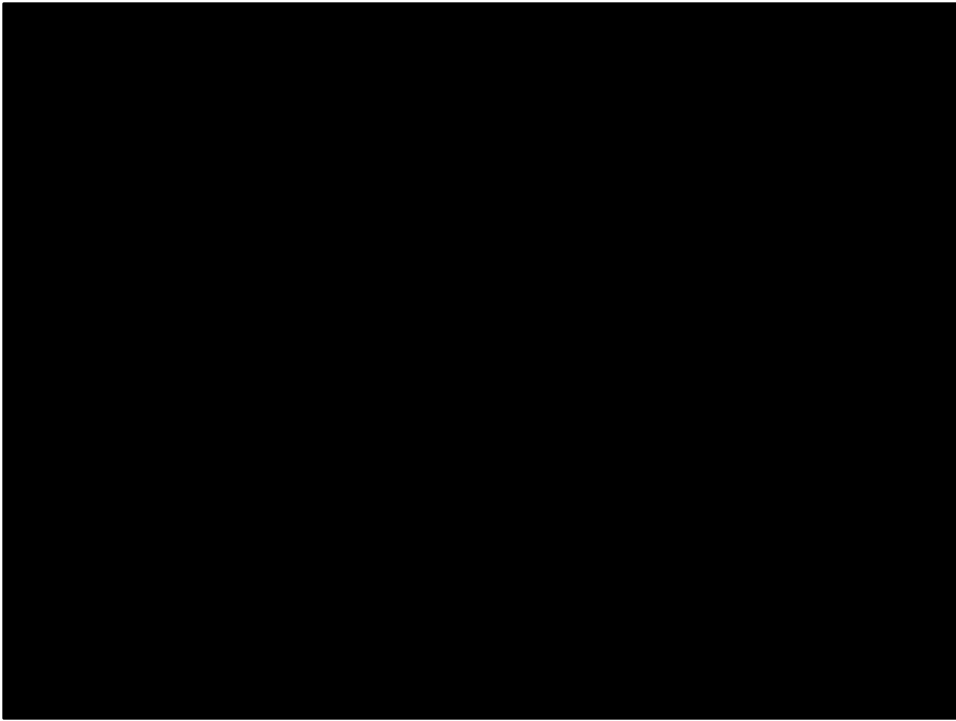
Fits all

=>

Can serve as a loss-less intermediate (canonical) format, e.g. for conversions.

3. XML format, with XML Schema (XSD)

Point number three. (Of three)



Now let's quickly remember what we were discussing about XML formats and XML Schema. (Such as the 3 main questions about a format)

/3. XML format, with XML Schema (XSD)

Advantage: standard parsing into objects

Advantage: standard binary serialisation

and more: some standard validation, detailed semantic annotation

End of point three (of three). The main advantages of XSD-based XML are summarised.

In the following notes...

By “Standard”/“automatic” I mean format-unspecific parsing/validation/serialisation, done by the standard XML/XSD-handling libraries. That means that nobody has to implement a specific parser/validator/serialiser.

By “Manually” I mean that format-specific parsers/validators/serialisers are necessary.

Note that:

“Standard parsing into objects” – let’s vaguely say that 90% of parsing is “automatic” with an average XML/XSD format.

Reason: XML is a tree, objects are a graph. The “10%” of the data that goes beyond the tree structure must be parsed “manually”.

“Some standard validation” – let’s super-vaguely say that 50% of validation is “automatic” with an average XML/XSD format.

(Although that depends on the way we look at it. Perhaps 90% is “automatic” with respect to pure validation of the “formatting” itself. But that does not have to mean that the data makes sense!)

Rules that can be expressed on the level of the XML Schema can be validated “automatically” (in case of Web services this can happen even before the requests

reach the service code). Beyond that, the validation (i.e. whether the data makes sense with respect to various criteria) must still be done “manually”.

BioXSD.org

Advantage: smooth workflows

Advantage: one format fits all

intermediate format for conversions

Advantage: standard parsing into objects

Advantage: standard binary serialisation

and more: some standard validation, detailed semantic annotation

BioXSD focusses on fulfilling/providing the discussed advantages, of great relevance each.

The yellow ones are those connected to being a common, rich format with a particular scope.

The green ones are connected to being an XML format well-defined by a dedicated machine-understandable XML Schema (XSD).

3



2

2



1

one

Extraaaaas ...

0

... Extras!

And here they're coming...

Goals of BioXSD:

- **Being an XSD-based XML format**
to complement plain-text/TSV and RDF formats
- **Filling the gap between specialised XSD-based exchange formats**
(such as SBML, MAGE-ML, PDBML, phyloXML, PSI-MI MIF, GCDML, GLYDE-II, ...)
- **Compatible with data binding libraries for all main programming languages**
- **As lightweight as possible, but fitting everyone**
- **Developed and maintained in an open but organised collaboration**
welcoming requests from the community
- **Detailed structure**
in-depth validation, semantic annotation (EDAM & more),
efficient compression (EXI)

Goals followed since the beginning of the development of BioXSD.
The high-level requirements

(For the other extras I had, it did not work for me to print them into this PDF with notes 😊)